



T.C
KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

Yazılım Laboratuvarı 1

Proje 2

HAZIRLAYANLAR:

220501001 Hatice Reyhan ÇALIŞKAN

220501009 Betül CANOL

210501007 Muhammed Yusuf KAYA

220501027 Evrim BAŞER

DERS SORUMLUSU:

Dr. Öğr. Üyesi Elif Pınar HACİBEYOęLU

TARİH

30.12.2024

1 GİRİŞ

1.1 Projenin amacı

Bu projenin amacı, öğrenci not tabloları, program çıktıları, ders çıktıları ve değerlendirme kriterleri arasında bir bağlantı kurarak, her bir öğrencinin ders ve program çıktılarındaki başarı oranlarını hesaplamak ve bunu uygun bir arayüz ile Excel tablosu şeklinde sunmaktır.

1.2 Gerçeklenmesi Beklenenler

- Crawler oluşturulması verilerin veritabanına kaydedilmesi.
- Program ve ders çıktıları arasında bir ilişki matrisi oluşturulması.
- Ders çıktıları ve değerlendirme kriterleri arasında bir ilişki matrisi oluşturulması.
- Her bir öğrenci için uygun ders çıktıları ve program çıktıları başarılarının hesaplanması.
- Uygun testlerin yapılması.
- Projenin raporlanması.

2 GEREKSİNİM ANALİZİ

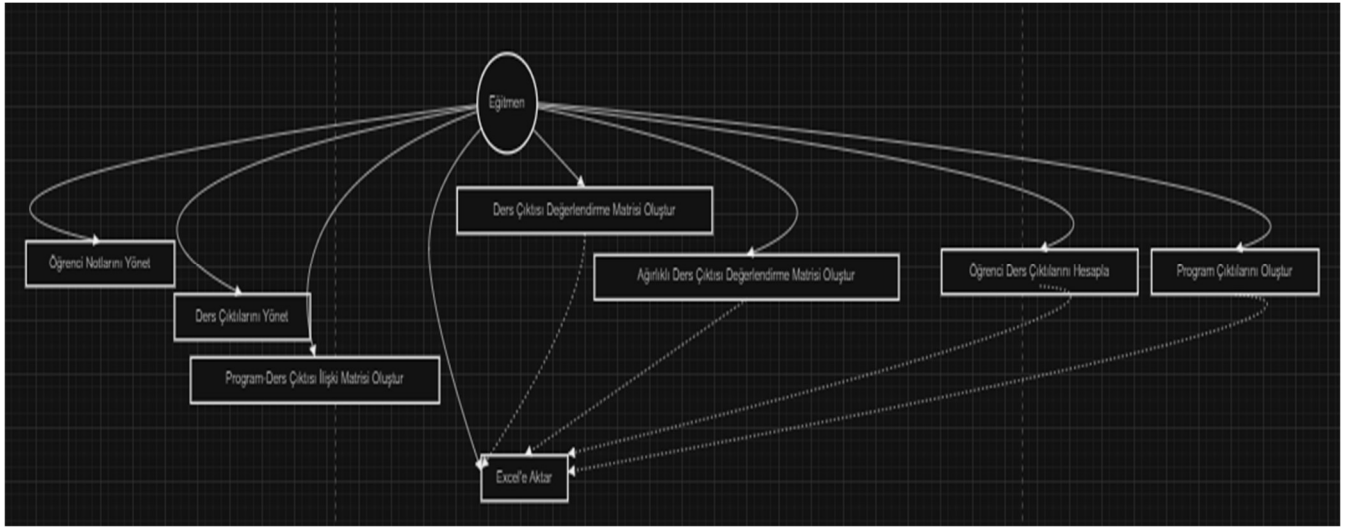
2.1 Arayüz gereksinimleri

- Kullanıcı dostu bir arayüzün tasarlanması.
- Notların manuel olarak girilebilmesi ve Excel dosyası okuma – yazma desteğinin olması.
- İstenilen tabloları görsel olarak görüntüleyebilme ve gerekli yerlerde düzenleyebilme.

2.2 Fonksiyonel gereksinimler

- Girdi olarak öğrenci not tablosu, program çıktıları/ders çıktıları matrisinin alınması.
- İlişki matrisleri oluşturma ve Excel’e kaydedebilme.
- Başarı oranlarının hesaplanıp tablo olarak sunulabilmesi.

2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

Kullanıcı Arayüzü Modülü:

- Tkinter ile oluşturulan basit bir GUI.
- Kullanıcıdan veri alır ve sonuçları görsel olarak sunar.

Veri İşleme Modülü:

- Program ve ders çıktıları ile değerlendirme kriterlerini işler.
- İlişki matrislerini hesaplar.

Excel Çıktı Modülü:

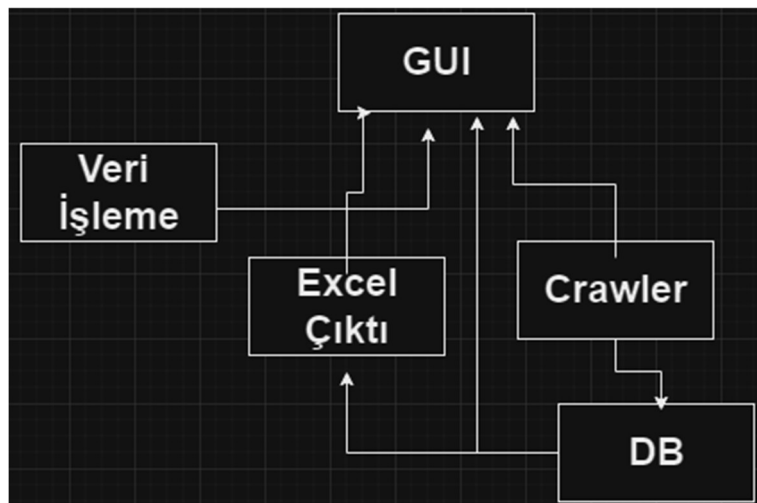
- pandas ve openpyxl kütüphaneleri ile Excel tablosu oluşturur.

Crawler Modülü:

- Web'den veri toplar ve SQLite veritabanlarına ekler.
- requests ve BeautifulSoup kullanılarak HTML tablolarını işler.

Veri Tabanı Yönetim Modülü:

- SQLite veritabanlarında işlemler gerçekleştirir.
- Verilerin güvenilir şekilde depolanmasını sağlar.

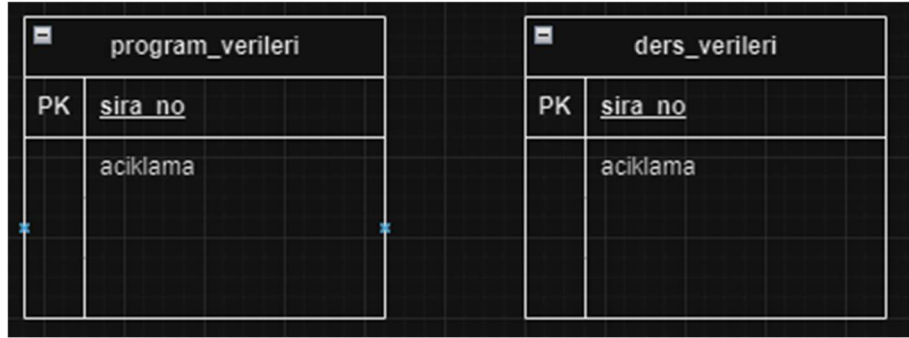


3.2 Kullanılacak teknolojiler

- Bu projede Python dili
- Projede kullanılan kütüphaneler;
 - BeautifulSoup: Crawler için
 - SQLite3: SQL veritabanı oluşturmak için
 - Request: HTTP istekleri yapmak için
 - Tkinter: Arayüz geliştirmek için
 - Pandas: Excel dosyaları ile çalışmak için
 - Openpyxl: Excel dosyasınında düzenleme yapmak için

3.3 Veri tabanı tasarımı

Bu projede veritabanı Crawler'ın istenilen websitelerinden çektiği verileri saklamak için kullanılmıştır. Ders Çıktıları ve Program Çıktıları tablolarından elde edilen veriler oluşturulan veritabanlarında sıra_no ve aciklama olarak kaydedilir. İki tablo arasında doğrudan bir ilişki bulunmamaktadır.



ER Diyagramı

3.4 Kullanıcı arayüzü tasarımı

Bu programın arayüzü tkinter kütüphanesi ile yapılmış olup notları ve ilişki matrislerini gösteren bir tablo kısmı ve kaydetme/ düzenleme excel'e aktarma vb. işlevleri gerçekleştirme amaçlı butonlar içermektedir.

OGRENCİ NOTLARI						
Ogrenci_No	Ödev1	Ödev2	Quiz	Vize	Final	Ortalama
220501001.0	100.0	100.0	90.0	100.0	100.0	98.0
220501009.0	70.0	60.0	80.0	50.0	60.0	64.0
220501002.0	10.0	50.0	33.0	50.0	60.0	40.6
220501003.0	100.0	30.0	54.0	58.0	100.0	68.4
220501004.0	76.0	25.0	70.0	50.0	30.0	50.2
220501005.0	70.0	76.0	90.0	65.0	100.0	80.2
220501006.0	55.0	44.0	10.0	63.0	40.0	42.40000000000000
220501007.0	80.0	58.0	100.0	70.0	50.0	71.6
220501011.0	0.0	0.0	0.0	0.0	0.0	0.0
220501111.0	0.0	0.0	0.0	0.0	0.0	0.0

KaydetSatır EkleSatır SilAğırlıkları Düzenle

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

crawler.py

- `add_data_to_sqlite`: Belirtilen URL'den tablo verisini SQLite veritabanına kaydetmek için gereken tüm işlemleri gerçekleştirir.
- `sqlite3`: Veritabanı bağlantısı kurar, SQL sorgularını çalıştırır ve işlemleri kalıcı hale getirir.
- `Requests`: Web sayfasından HTTP isteğiyle HTML içeriğini alır.
- `BeautifulSoup`: HTML içeriğini analiz ederek tablo gibi elemanları ayırır.

tablo-not.py

- `ProgramMatrisUygulamasi`: Kullanıcıların program çıktıları ve öğrenci notlarını düzenlemelerine olanak tanıyan bir tkinter tabanlı arayüz.
- `create_tabs(self)`: Uygulama sekmelerini oluşturur ve yükler.
- `create_student_notes_tab(self, tab_frame, excel_file)`:
 - Öğrenci notlarını düzenlemek için bir sekme oluşturur.
 - Ağırlıklı ortalamaları hesaplar ve tabloyu düzenler.
- `edit-weights(self, df, refresh_callback)`:
 - Kullanıcının ödev ağırlıklarını düzenlemesine olanak tanır.
- `load_excel_to_tab(self, tab_frame, excel_file)`: Genel amaçlı Excel dosyalarını yüklemek ve görüntülemek için bir sekme oluşturur.
- `save_excel(self, df, excel_file)`: Tabloyu belirtilen Excel dosyasına kaydeder.
- `calculate_weighted_average(self, df)`: Her satır için ağırlıklı ortalamayı hesaplar.

tablo-1.py

- `get_tables`: Veritabanından ders veya program çıktısı tablolarını getirir.
- `load_data`: Veritabanlarının varlığını kontrol eder ve matrisi oluşturur.
- `update_matrix`: Matris çerçevesini temizler ve verilerle yeniden oluşturur.
- `calculate_relation`: Belirli bir satırın ortalama ilişki değerini

hesaplar.

- `on_entry_change`: Giriş alanı değiştiğinde ilişki değerlerini yeniden hesaplar.
- `update_relation_labels`: Tüm satırlardaki ilişki değerlerini günceller.
- `export_to_excel`: Matris verilerini bir Excel dosyasına kaydeder.

tablo-2-3.py

- `validate_weight(self, P)`: Kullanıcı tarafından girilen ağırlıkların geçerliliğini kontrol eder.
- `validate_matrix_value(self, P)`: Matrisin değerlerinin geçerli olup olmadığını kontrol eder.
- `create_weights_row(self)`: Ders ağırlıkları girebilmek için bir satır oluşturur.
- `check_weights_sum(self, event=None)`: Ağırlıkların toplamını kontrol eder ve toplamın 100 olup olmadığını kullanıcıya bildirir. Böylece yanlış hesaplamaların önüne geçilir.
- `get_course_outputs(self)`: Veritabanından her ders çıktısının sırası ve açıklamasını getirir.
- `on_entry_focus_in(self, entry)`: Kullanıcı bir giriş kutusuna tıkladığında, eğer kutuda "0" yazıyorsa, bu değer silinmesini sağlar.
- `create_matrix(self)`: Ders çıktıları ve değerlendirme türleri ile ilgili matrisi oluşturur.
- `create_weighted_matrix(self)`: Ağırlıklı matrisin oluşturulmasından sorumludur.
- `update_weighted_matrix(self, event=None)`: Ağırlıklı matrisin güncellenmesinden sorumludur.
- `save_to_excel(self)`: Kullanıcı tarafından girilen verileri excel'e kaydeder.

tablo-4.py

- `load_data(self)`: Excel dosyalarından öğrenci notları ve ders çıktıları verilerini yükler, veritabanından ders çıktıları alır.
- `get_ders_ciktilari_from_db(self)`: Ders çıktıları veritabanından alınır.
- `calculate_for_all_students(self)`: Tüm öğrenciler için hesaplama yapar ve her öğrenciye özel tablonun görüntülenmesini sağlar.

- `get_ders_ciktilari_from_db(self)`: Ders çıktıları veritabanından alınır.
- `calculate_for_all_students(self)`: Tüm öğrenciler için hesaplama yapar ve her öğrenciye özel tablonun görüntülenmesini sağlar.
- `calculate_and_display(self, tree, student)`: Her öğrenci için ders çıktıları hesaplamalarını yapar ve sonucu ilgili tabloya ekler.
- `save_all_to_excel(self)`: Hesaplanan tüm tabloları Excel dosyalarına kaydeder.

tablo-5.py

- `safe_float_convert(self, value)`: Girilen bir değeri güvenli bir şekilde float türüne dönüştürür.
- `get_student_numbers(self)`: `ogrenci_notlari.xlsx` dosyasını okuyarak öğrencilerin numaralarını alır.
- `read_tablo1(self)`: `tablo_1.xlsx` dosyasından ilişki değerlerini ve numeric değerleri okur.
- `process_all_students(self)`: Tüm öğrenciler için "Tablo 5" dosyalarını oluşturur.
- `create_tablo_5(self, tablo4_filename, student_no, tablo1_values, iliski_degerleri)`: Tablo 5 Excel dosyasını oluştur.

4.2 Görev dağılımı

Crawler, yazılımın iç mimarisi, arayüz ve veritabanı yönetimi görevleri grup içinde eşit olarak paylaştırılmıştır.

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Crawler oluşturulurken websitesinden verileri doğru çekememek.
 - Githubdan örnek kodlara bakıldı.
- Tablolar birbiri ile ilişkilendirilirken zorluklar yaşandı.
 - İlgili kaynaklar ve örnek kodlar ile probleme çözüm bulundu.
- Arayüzde Excel formatının gösterilememesi.
 - Arayüz kütüphanesine ait tutorial videolar izlendi.

4.4 Proje isterlerine göre eksik yönler

- Proje isterlerine göre herhangi bir eksik yön bulunmamaktadır.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci

Programın arayüze sahip olmasından dolayı test kodu bulunmamaktadır. Kullanıcı girişleri ile test edilmiştir.

5.2 Yazılımın doğrulanması

Yapılan testler doğrultusunda kod tam doğrulukta çalışmaktadır.

6 GITHUB LİNKLERİ

- https://github.com/ReyhanClskn/Lab_1_Proje_2
- <https://github.com/yusufky56>
- <https://github.com/BetulCnl>
- <https://github.com/evrimbaserr>